

Colcon in-container

Say bye to the “it builds on my machine”

ROSCon 2023

Guillaume Beuzeboc, robotics engineer @



Canonical

Say bye to “It builds on my machine”

We can all relate to this story,

- You want to use this shiny new package
- Run rosdep and try to build
- It fails
- It is missing a dependency
- Mix between build and run

```
Starting >>> newpackage
--- stderr: turtlebot4_base In file /home/ubuntu/ws/src/newpackage/include/newpackage/gpio_interface.hpp:23:10: fatal
error: gpiod.h: No such file or directory
   23 | #include <gpiod.h>
      |           ^~~~~~
compilation terminated.
```

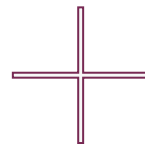


What is it?

In-container - a new colcon extension

What is colcon in-container

- Colcon verb extensions
- Run your colcon command inside a fresh and isolated ROS 2 env
- Use your packages definitions to install required dependencies
- Transfers the results back to the host before removing the environment

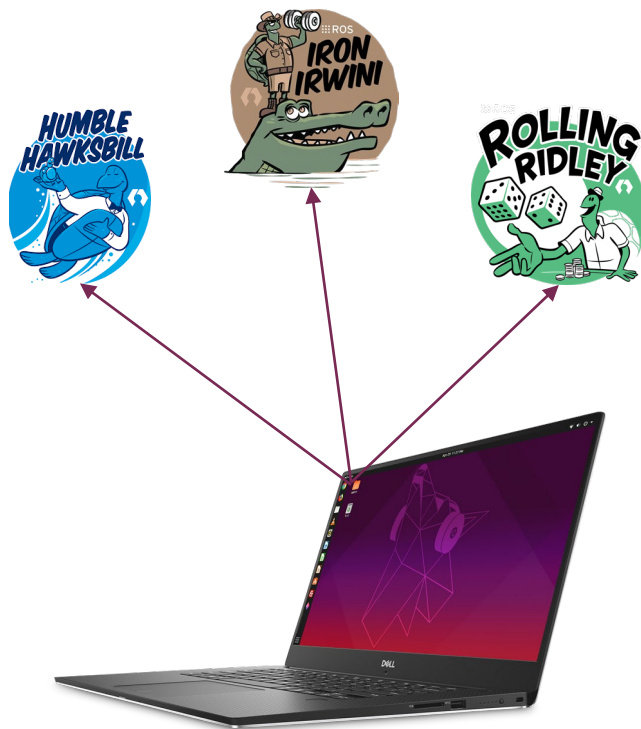


Canonical
LXD

In-container - a new colcon extension

Colcon in-container use cases

- Build and test a ROS 2 package for any ROS 2 distro
- Isolate build and tests
- Validate your package.xml
- Prepare for a release
- Prepare for packaging



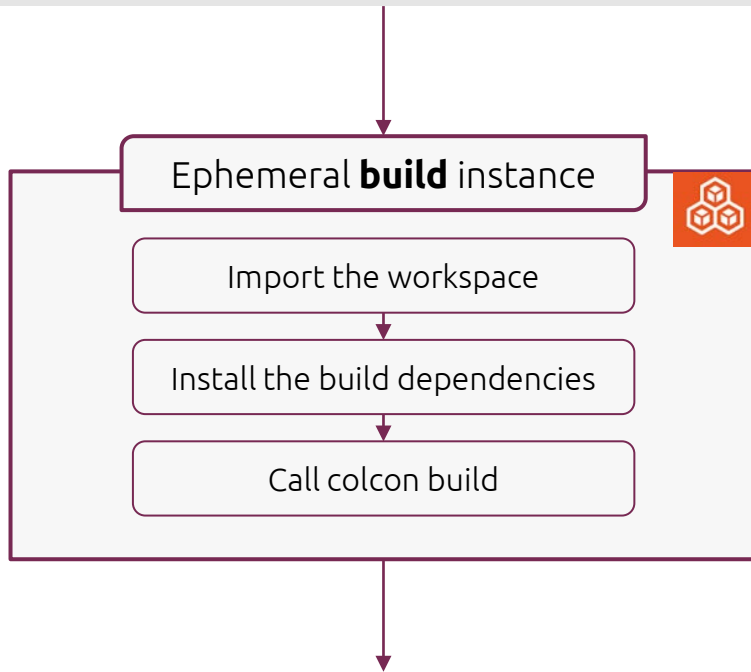


What does it do?

In-container - a new colcon extension

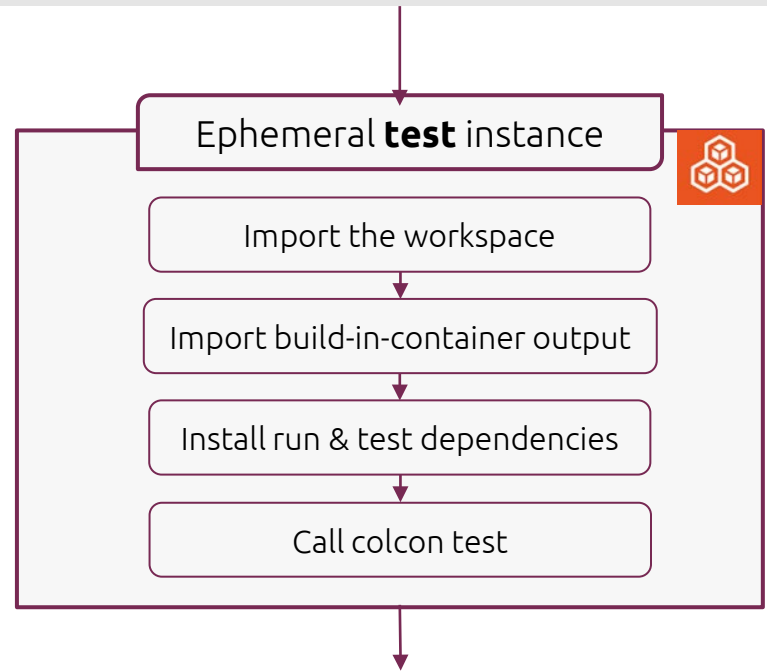
The workflow

```
$ colcon build-in-container --ros-distro humble
```



Results are downloaded on the host

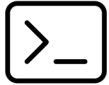
```
$ colcon test-in-container --ros-distro humble
```



Results are downloaded on the host

Usage

Build-in-container



```
$ colcon build-in-container --ros-distro <ROS_DISTRO>
```

```
INFO:colcon.colcon-in-container:Downloading the image then creating the LXD instance
```

```
INFO:colcon.colcon-in-container:Waiting for ROS 2 to be installed
```

```
INFO:colcon.colcon-in-container.instance:....
```

```
[...]
```

```
INFO:colcon.colcon-in-container.instance: Summary: 1 package finished
```

```
INFO:colcon.colcon-in-container:downloading /ws/install on host
```

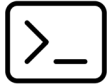
```
INFO:colcon.colcon-in-container:downloading /ws/build on host
```



```
build_in_container/ install_in_container/ log/ src/
```


Usage

Test-in-container



```
$ colcon test-in-container --ros-distro <ROS_DISTRO>
```

```
INFO:colcon.colcon-in-container:Downloading the image then creating the LXD instance  
INFO:colcon.colcon-in-container:Waiting for ROS 2 to be installed  
INFO:colcon.colcon-in-container.instance:....  
[...]
```



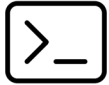
```
INFO:colcon.colcon-in-container.instance: Summary: 1 package finished  
INFO:colcon.colcon-in-container:downloading /ws/test_results on host
```



```
test_results_in_container/ build_in_container/ install_in_container/ log/ src/
```

Usage

Instance introspection



```
$ colcon build-in-container --ros-distro <ROS_DISTRO> --debug/--shell-after
```

--debug

Shell in the container in case of an error

--shell-after

Shell in the container at the end of the command or in case of an error

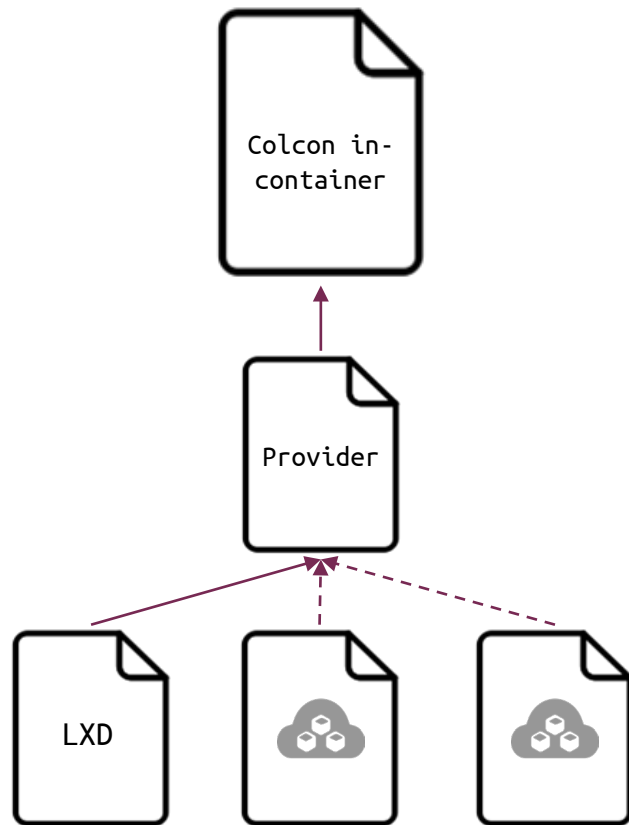
Backend

Adaptable to multiple backend

- Ephemeral environments are provided by an LXD container
- The backend provider is interfaced so additional one could be added in the future



Canonical
LXD





What is next?

Open Source

- `pip3 install -U git+https://github.com/canonical/colcon-in-container`
- Available for Humble, Iron and Rolling

What's cooking

- Integration of release workflow
- Implement more colcon verbs
- Additional providers (e.g. Multipass)



canonical / **colcon-in-container**



Thank you. Questions?